INTERNATIONAL STANDARD

ISO/IEC 15946-5

Third edition
2022-02

Information security — Cryptographic techniques based on elliptic curves —

Part 5: Elliptic curve generation

Sécurité de l'information — Techniques cryptographiques fondées sur les courbes elliptiques —
Partie 5: Génération de courbes elliptiques

Reference number
ISO/IEC 15946-5:2022(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee ISO/IEC JTC 1/SC 27, *Information security, cybersecurity and privacy protection*.

This third edition cancels and replaces the second edition (ISO/IEC 15946-5:2017), which has been technically revised.

The main changes compared to the previous edition are as follows:

— BLS curves have been added to Clause 7;

— security background for pairing-friendly curves has been added to Annex B, including the exTNFS attack that affects the security of numerical examples of BN curves;

— except for BN curves, all other curves have been moved to Annex C;

— associated numerical examples (Annex D) and properties (Annex E) have been updated.

A list of all parts in the ISO/IEC 15946 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Some of the most interesting alternatives to the RSA and $F(p)$ based systems are cryptosystems based on elliptic curves defined over finite fields. The concept of an elliptic curve based public-key cryptosystem is rather simple.

— Every elliptic curve over a finite field is endowed with an addition operation "+", under which it forms a finite abelian group.

— The group law on elliptic curves extends in a natural way to a "discrete exponentiation" on the point group of the elliptic curve.

— Based on the discrete exponentiation on an elliptic curve, one can easily derive elliptic curve analogues of the well-known public-key schemes of Diffie-Hellman and ElGamal type.

The security of such a public-key system depends on the difficulty of determining discrete logarithms in the group of points of an elliptic curve. With current knowledge, this problem is much harder than the factorization of integers or the computation of discrete logarithms in a finite field. Indeed, since Miller and Koblitz independently suggested the use of elliptic curves for public-key cryptographic systems in 1985, the elliptic curve discrete logarithm problem has only been shown to be solvable in certain specific and easily recognizable cases. There has been no substantial progress in finding an efficient method for solving the elliptic curve discrete logarithm problem on arbitrary elliptic curves. Thus, it is possible for elliptic curve based public-key systems to use much shorter parameters than the RSA system or the classical discrete logarithm-based systems that make use of the multiplicative group of a finite field. This yields significantly shorter digital signatures and system parameters.

The purpose of this document is to meet the increasing interest in elliptic curve based public-key technology by describing elliptic curve generation methods to support key management, encryption and digital signatures based on an elliptic curve.

Information security — Cryptographic techniques based on elliptic curves —

Part 5: Elliptic curve generation

1 Scope

The ISO/IEC 15946 series specifies public-key cryptographic techniques based on elliptic curves described in ISO/IEC 15946-1.

This document defines elliptic curve generation techniques useful for implementing the elliptic curve based mechanisms defined in ISO/IEC 29192-4, ISO/IEC 9796-3, ISO/IEC 11770-3, ISO/IEC 14888-3, ISO/IEC 18033-2 and ISO/IEC 18033-5.

This document is applicable to cryptographic techniques based on elliptic curves defined over finite fields of prime power order (including the special cases of prime order and characteristic two). This document is not applicable to the representation of elements of the underlying finite field (i.e. which basis is used).

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15946-1, *Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 1: General*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 15946-1 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp;

— IEC Electropedia: available at https://www.electropedia.org/.

3.1
cryptographic hash function
function that maps octet strings of any length to octet strings of fixed length, such that it is computationally infeasible to find correlations between inputs and outputs, and such that given one part of the output, but not the input, it is computationally infeasible to predict any bit of the remaining output

[SOURCE: ISO/IEC 18033-2:2006, 3.11, modified — Deleted the last phrase, "The precise security requirements depend on the application.]

3.2
definition field of an elliptic curve
field that includes all the coefficients of the formula describing an elliptic curve

3.3
hash-function
function which maps strings of bits of variable (but usually upper bounded) length to fixed-length strings of bits, satisfying the following two properties:

— for a given output, it is computationally infeasible to find an input which maps to this output;

— for a given input, it is computationally infeasible to find a second input which maps to the same output.

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment. Refer to ISO/IEC 10118-1:2016, Annex C.

[SOURCE: ISO/IEC 10118-1:2016, 3.4]

3.4
nearly prime number
positive integer, $n = m \cdot r$, where m is a large prime number and r is a small *smooth integer* (3.6)

Note 1 to entry: The meaning of the terms large and small prime numbers is dependent on the application and is based on bounds determined by the designer.

3.5
order of an elliptic curve $E(F)$
number of points on an elliptic curve, E, defined over a finite field, F

3.6
smooth integer
integer, r, whose prime factors are all small, i.e. less than some defined bound

4 Symbols and conversion functions

4.1 Symbols

0_F	zero element in a field F
a, b	elliptic curve parameters
B	embedding degree, the smallest B such that number $\#E(F(q))$ is a divisor of q^B-1
E	elliptic curve, given by an equation of the form $Y^2 = X^3 + aX + b$ over the field $F(p^m)$ for $p > 3$, by an equation of the form $Y^2 + XY = X^3 + aX^2 + b$ over the field $F(2^m)$, or by an equation of the form $Y^2 = X^3 + aX^2 + b$ over the field $F(3^m)$, together with an extra point O_E referred to as the point at infinity; the curve is denoted by $E/F(p^m)$, $E/F(2^m)$, or $E/F(3^m)$, respectively
F	finite field
$F(p)^*$	multiplicative group of $F(p)$
L, m	positive integers
mod	modulo, a mod b means a residue of a modulo b for integers a and b (b is positive)
N	number of points on an elliptic curve E over $F(q)$, $\#E(F(q))$
n	prime divisor of $\#E(F(q))$
O_E	elliptic curve point at infinity
p	prime number

q	prime power, p^m for some prime p and some integer $m \geq 1$
r	cofactor, that is $\#E(F(q)) = rn$
u	positive integer
v	positive integer
λ	positive integer
$v(i)$	polynomial of i
$\#E(F(q))$	order of an elliptic curve $E(F(q))$
$\lceil x \rceil$	smallest integer greater than or equal to the real number x
$\lfloor x \rfloor$	largest integer smaller than or equal to the real number x
$\cup$	or
$\|\|$	concatenation
$\in$	element is included in a set

4.2 Conversion functions

BS2IP	bit string to integer conversion primitive
BS2OSP	bit string to octet string conversion primitive
I2BSP	integer to bit string conversion primitive
OS2FEP	octet string to finite field element conversion primitive

NOTE Refer to ISO/IEC 15946-1:2016, Clause 7 for detailed conversion functions.

5 Conventions for elliptic curves

5.1 Definitions of elliptic curves

5.1.1 Elliptic curves over $F(p^m)$

Let $F(p^m)$ be a definition field of an elliptic curve, where $F(p^m)$ is a finite field with a prime $p > 3$ and a positive integer m. In this document, it is assumed that E is described by a "short (affine) Weierstrass equation", that is an equation of type

$$Y^2 = X^3 + aX + b \text{ with } a, b \in F(p^m)$$

such that $4a^3 + 27b^2 \neq 0_F$ holds in $F(p^m)$.

NOTE 1 The above curve with $4a^3 + 27b^2 = 0_F$ is called a singular curve, which is not an elliptic curve.

The set of $F(p^m)$-valued points of E is given by Formula (1):

$$E(F(p^m)) = \{Q = (x_Q, y_Q) \in F(p^m) \times F(p^m) | {y_Q}^2 = {x_Q}^3 + ax_Q + b\} \cup \{O_E\} \tag{1}$$

where O_E is an extra point referred to as the point at infinity of E.

NOTE 2 In applications not based on a pairing, $E/F(p)$ or $E/F(2^m)$ is preferable from an efficiency point of view. In applications that use a pairing, $E/F(p)$ is preferable from an efficiency point of view.

5.1.2 Elliptic curves over $F(2^m)$

Let $F(2^m)$ for $m \geq 1$ be a definition field of an elliptic curve. In this document, it is assumed that E is described by an equation of the type

$$Y^2 + XY = X^3 + aX^2 + b \text{ with } a, b \in F(2^m)$$

such that $b \neq 0_F$ holds in $F(2^m)$.

For cryptographic use, m shall be a prime to prevent certain kinds of attacks on the cryptosystem.

NOTE The above curve with $b = 0_F$ is called a singular curve, which is not an elliptic curve.

The set of $F(2^m)$-valued points of E is given by Formula (2):

$$E(F(2^m)) = \{Q = (x_Q, y_Q) \in F(2^m) \times F(2^m) | {y_Q}^2 + x_Q y_Q = {x_Q}^3 + a{x_Q}^2 + b\} \cup \{O_E\} \tag{2}$$

where O_E is an extra point referred to as the point at infinity of E.

5.1.3 Elliptic curves over $F(3^m)$

Let $F(3^m)$ be a definition field of an elliptic curve, where $F(3^m)$ is a finite field with a positive integer m. In this document, it is assumed that E is described by an equation of the type

$$Y^2 = X^3 + aX^2 + b \text{ with } a, b \in F(3^m)$$

such that $a, b \neq 0_F$ holds in $F(3^m)$.

NOTE The above curve with a or $b = 0_F$ is called a singular curve, which is not an elliptic curve.

The set of $F(3^m)$-valued points of E is given by Formula (3):

$$E(F(3^m)) = \{Q = (x_Q, y_Q) \in F(3^m) \times F(3^m) | {y_Q}^2 = {x_Q}^3 + a{x_Q}^2 + b\} \cup \{O_E\} \tag{3}$$

where O_E is an extra point referred to as the point at infinity of E.

5.2 Group law on elliptic curves

Elliptic curves are endowed with the addition operation $+: E \times E \to E$, defining for each pair (Q_1, Q_2) of points on E a third point $Q_1 + Q_2$. With respect to this addition, E is an abelian group with identity element O_E. The kth multiple of Q is given as kQ, where $kQ = Q + \ldots + Q$ (k summands) if $k > 0$, $kQ = (-k)(-Q)$ if $k < 0$, and $kQ = O_E$ if $k = 0$. The smallest positive k with $kQ = O_E$ is called the order of Q.

In order to use an elliptic curve for a cryptosystem, it is necessary to compute group law on an elliptic curve. ISO/IEC 15946-1 shall be referred to for methods of group law on elliptic curves.

6 Framework for elliptic curve generation

6.1 Trust in elliptic curve

If an elliptic curve is poorly or maliciously generated, this can enable an adversary to break a cryptosystem that uses it. There are a number of ways in which a user can obtain trust in the provenance of an elliptic curve, including the following:

— The curve can be obtained from an impartial trusted source (e.g. an international or national standard).

— The curve can be generated and/or verified by a trusted third party.

— The curve can be generated and/or verified by the user.

NOTE 1 Refer to Annex A for background information on elliptic curves.

NOTE 2 Refer to Annex B for background information on elliptic curve cryptography.

6.2 Overview of elliptic curve generation

There are three main ways to generate elliptic curves:

— by applying the order of counting algorithms to a (pseudo-)randomly chosen elliptic curve as specified in Clause 7;

— by applying a complex multiplication method as specified in Clause 8;

— by lifting an elliptic curve over a small finite field over a reasonably large field as specified in Clause 9.

NOTE 1 Refer to Annex A for background information on elliptic curves.

NOTE 2 Refer to Annex B for background information on elliptic curve cryptography.

7 Verifiably pseudo-random elliptic curve generation

7.1 General

The generation of verifiably pseudo-random elliptic curves focuses on curves over prime and binary fields.

7.2 Constructing verifiably pseudo-random elliptic curves (prime case)

7.2.1 Construction algorithm

The following algorithm produces a set of elliptic curve parameters over a field $F(p)$ selected (pseudo-) randomly from the curves of appropriate order, along with sufficient information for others to verify that the curve was indeed chosen pseudo-randomly.

NOTE 1 The algorithm is consistent with IEEE P1363-2000.

NOTE 2 Methods of choosing a prime number p (pseudo) randomly are described in ISO/IEC 18032.

It is assumed that the following quantities have been chosen:

— a lower bound, $n_{\min}$, for the order of the base point;

— a cryptographic hash function, H, with output length L_{Hash} bits;

— the bit length, L, of inputs to H, satisfying $L \geq L_{\text{Hash}}$.

The following notation is adopted below:

— $v = \lceil \log_2 p \rceil$;

— $s = \lfloor (v - 1)/L_{\text{Hash}} \rfloor$;

— $w = v - sL_{\text{Hash}} - 1$.

Input: a prime number p; lower bound $n_{\min}$ for n; a trial division bound $l_{\max}$.

Output: a bit string X; EC parameters a, b, n, and G.

a) Choose an arbitrary bit string X of bit length L.

b) Compute $h = \text{H}(X)$.

c) Let W_0 be the bit string obtained by taking the w rightmost bits of h.

d) Let $Z = \text{BS2IP}(X)$.

e) For i from 1 to s, do the following:

   1) Let $X_i = \text{I2BSP}(Z + i \bmod 2^L)$.

   2) Compute $W_i = \text{H}(X_i)$.

f) Let $W = W_0 \,||\, W_1 \,||\, \ldots \,||\, W_s$.

g) Let $c = \text{OS2FEP}[\text{BS2OSP}(W)]$.

h) If $c = 0_F$ or $4c + 27 = 0_F$, then go to step a).

i) Choose finite field elements a, $b \in F(p)$ such that $b \neq 0_F$ and $cb^2 - a^3 = 0_F$. Choosing $a = b = c$ guarantees the conditions hold, and this choice is recommended.

   NOTE 3 It is possible that choosing $a = b = c$ is not optimal from a performance perspective.

   NOTE 4 If the default values are chosen as suggested, the randomness of the generated curve is explicitly guaranteed.

j) Compute the order $\#E(F(p))$ of the elliptic curve E over $F(p)$ given by $y^2 = x^3 + ax + b$.

k) Test whether $\#E(F(p))$ is a nearly prime number using the algorithm specified in 7.2.2. If so, the output of the algorithm specified in 7.2.2 consists of integers r, n. If not, then go to step a).

   NOTE 5 The necessity of near primality is described in B.2.2

l) Check if $E(F(p))$ satisfies the MOV-condition specified in B.2.3, that is the smallest integer B such that n divides $q^B - 1$ ensures the desirable security level. If not, then go to step a).

m) If $\#E(F(p)) = p$, then go to step a).

   NOTE 6 This check is performed in order to protect against the attack specified in B.2.2.

n) Test whether the prime divisor n satisfies the condition described in B.2.4 for cryptosystems based on ECDLP, ECDHP, or BDHP with auxiliary inputs as in B.1.5. If not, then go to step a).

o) Generate a point G on E of order n using the algorithm specified in 7.2.3.

p) Output X, a, b, n, G.

NOTE 7 Methods to compute the order $\#E(F(p))$ are described in References [12], [31] and [32].

7.2.2 Test for near primality

Given a lower bound n_{min} and a trial division bound l_{max}, the following procedures test $N = \#E(F(p))$ for near primality.

Input: positive integers N, l_{max}, and n_{min}.

Output: if N is nearly prime, output a prime n with $n_{min} \leq n$ and a smooth integer r such that $N = rn$. If N is not nearly prime, output the message "not nearly prime".

a) Set $n = N$, $r = 1$.

b) For l from 2 to l_{max}, do the following:

   1) If l is composite, then go to step 3).

   2) While (l divides n)

      i) Set $n = n/l$ and $r = rl$.

      ii) If $n < n_{min}$, then output "not nearly prime" and stop.

   3) Next l.

c) Test n for primality.

d) If n is prime, then output r and n and stop.

e) Output "not nearly prime".

NOTE Methods to test for primality are described in ISO/IEC 18032 and Reference [11].

7.2.3 Finding a point of large prime order

If the order $\#E(F(q))$ of an elliptic curve E is nearly prime, the following algorithm efficiently produces a random point in $E(F(q))$ whose order is the large prime factor n of $\#E(F(q)) = rn$.

Input: an elliptic curve E over the field $F(q)$, a prime n, and a positive integer r not divisible by n.

Output: if $\#E(F(q)) = rn$, a point G on E of order n; if not, the message "wrong order."

a) Generate a random point P (not O_E) on E.

b) Set $G = rP$.

c) If $G = O_E$, then go to step a).

d) Set $Q = nG$.

e) If $Q \neq O_E$, then output "wrong order" and stop.

f) Output G.

7.2.4 Verification of elliptic curve pseudo-randomness

The following algorithm determines whether or not an elliptic curve over $F(p)$ was generated using the method of 7.2.1. The quantities L_{Hash}, L, v, s, and w, and the hash-function H, are as in 7.2.1.

Input: a bit string X of length L, EC parameters $q = p$, a, b, n, and $G = (x_G, y_G)$, and a positive integer n_{min}.

Output: "True" or "False".

a) Compute $h = H(X)$.